# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-249910
(43)Date of publication of application : 17.09.1999

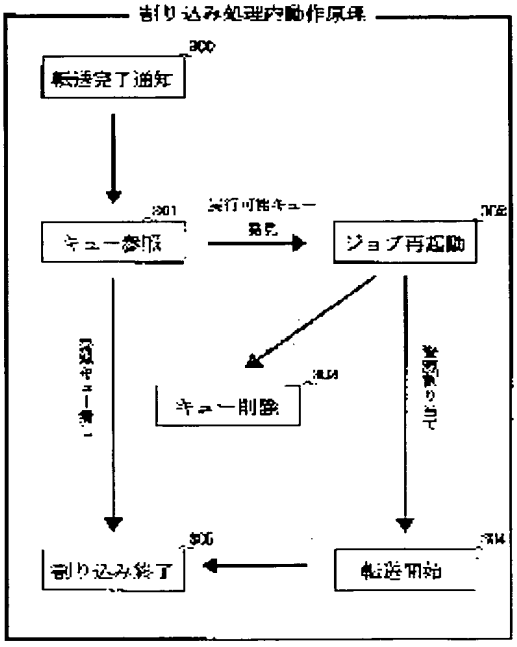
(51)Int.Cl. G06F 9/46
G06F 9/46
G06F 3/06

(21)Application number : 10-046741
(22)Date of filing : 27.02.1998

(71)Applicant : HITACHI LTD
(72)Inventor : KOBAYASHI NAOTAKA
NAGASE NORIKAZU
OGASAWARA YUTAKA
KUWABARA HIROSHI

## (54) STORAGE SUBSYSTEM AND DATA TRANSFER CONTROL METHOD

(57)Abstract:
PROBLEM TO BE SOLVED: To attain the effective use of a processor by reactivating only a job that is registered in a queue while referring to a queue table and starting the transfer of data when the jobs including the reactivation through the allocation of resources are executed.
SOLUTION: The reference is made to a queue table which is used for the queue registering (301). In this step 301, the registered queues are searched and an executable queue is detected. The detected executable queue is equal to a job queue that is registered after the current job queue which received the end notification and also equal to a queue that is logically connected to its preceding queue on the queue table. The transfer job of the queue is set in a transfer end wait state when the queue is registered and only this job is reactivated (302). Then the activated job is immediately deleted (303). Then the transfer resources are allocated to the activated job and this job is instructed to start the transfer of data (304). The interrupt processing is ended after the transfer of data is started (305).




LEGAL STATUS

[Date of request for examination] 05.02.2003

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

**特開平11－249910**

(43)公開日　平成11年(1999)９月17日

| (51)Int.Cl.[6] | | 識別記号 | FI | | |
|---|---|---|---|---|---|
| Ｇ０６Ｆ | 9/46 | ３４０ | Ｇ０６Ｆ | 9/46 | ３４０Ａ<br>Ｃ |
| | 3/06 | ３０１ | | 3/06 | ３０１Ｍ |

審査請求　未請求　請求項の数２　ＯＬ　（全 11 頁）

(21)出願番号　特願平10－46741

(22)出願日　平成10年(1998)２月27日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地

(72)発明者　小林　直孝
神奈川県小田原市国府津2880番地　株式会社日立製作所ストレージシステム事業部内

(72)発明者　長瀬　則和
神奈川県小田原市国府津2880番地　株式会社日立製作所ストレージシステム事業部内

(72)発明者　小笠原　裕
神奈川県小田原市国府津2880番地　株式会社日立製作所ストレージシステム事業部内

(74)代理人　弁理士　小川　勝男

最終頁に続く

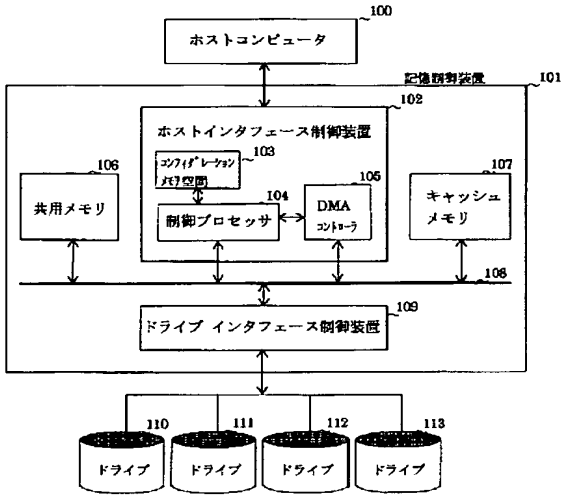
(54)【発明の名称】　記憶サブシステム及びデータ転送制御方法

(57)【要約】

【課題】複数のホストコンピュータと記憶装置（ストレージ）間でチャネルを介したデータ転送制御に関し、頻繁なデータ入出力要求からなる多重転送ジョブを、効率良く管理し、円滑な資源展開を可能とし、プロセッサの付加を軽減する。

【解決手段】ジョブ起動時に転送資源が空いているかどうかをチェックし、空いていれば資源を割り当てて転送を開始する。もし、空いていない場合は転送資源待ちにする代わりに転送完了待ちとしてキューテーブルに次回転送の予約をする。転送完了待ちとすることでジョブでは再起動を行う必要はない。そして、再起動～資源割り当ての作業を現在転送中のジョブが割り込みの完了通知を行うタイミングで行うものとする。この時、キューテーブルを参照してキュー登録されている１ジョブのみの再起動を行い転送を開始する。素早く目的のジョブが発見できるように、キュー構造にはNEXTキューのポインタ情報をもたせる。

図１



1

【特許請求の範囲】
【請求項１】データを格納する複数の記憶装置と、前記記憶装置とホストコンピュータとのデータ転送を制御する記憶制御装置とを有する記憶サブシステムにおいて、
前記記憶制御装置は、データ入出力要求があったときはジョブを起動し、データ転送が実行されないジョブを管理し、管理するジョブを連続して実行するよう制御する制御部と、
前記ジョブと連続して実行される前記ジョブの順序とを登録する手段とを有することを特徴とする記憶サブシステム。
【請求項２】データを格納する複数の記憶装置とホストコンピュータとのデータ転送を制御する記憶制御装置のデータ転送制御装置であって、
データ入出力要求があったときはジョブを起動し、データ転送が実行されないジョブはキュー登録し、実行中のジョブが終了した場合はキュー登録されたジョブを起動することを特徴とするデータ転送制御装置。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、記憶装置（ストレージ）を有する記憶サブシステムに関し、特にホストコンピュータと記憶装置（ストレージ）間での記憶制御装置を介したデータ転送の際に多重転送ジョブを効率良く実行する技術に関する。
【０００２】
【従来の技術】従来の技術では、ジョブは〈１〉入力データの準備〈２〉起動〈３〉実行待ちの制御〈４〉資源の割り当て〈５〉実行〈６〉資源の解放〈７〉出力〈８〉終了までを一連の流れとして制御し、これが一般的なオペレーティングシステムによるジョブ管理方式である。、転送ジョブにおいても同様であり、複数の転送ジョブが重なると、実行されるジョブは１つだけなので残りのジョブは転送資源解放待ちの待ち状態となり、転送資源解放時に再び〈２〉～〈５〉の処理が繰り返される。つまりジョブの多重度が増すと優先順位の低いジョブはなかなか資源が割り当てられずに、何度も状態遷移を繰り返すためプロセッサを不要に使用してしまう。また転送終了から次の転送起動までの間に優先順位の低いジョブを起動するのは時間のロスである。
【０００３】これと類似のプロセス（タスク）の管理において、同様の問題を解決するために情報処理システム内で動作するプロセス（タスク）を一括して管理し、システムの処理能力の向上及びシステムの機能拡大に対応可能とするプロセス管理方式が特開平7-175671号に開示されている。
【０００４】
【発明が解決しようとする課題】特開平7-175671号に開示された技術は、プロセス（タスク）の管理であるので、ジョブにおける〈５〉実行に相当する部分に限定されていた。また、この技術は一旦停止したプロセスを再起動する際に、他のプロセス（タスク）と識別するために任意の識別情報を使用しすべてのプロセス（タスク）を管理することで不要なプロセスの起動を抑止している。本発明ではジョブ管理における不要な状態遷移を抑止するが、転送ジョブが重なった場合の転送資源開放時においてジョブの中で再起動を必要としない方式及び記憶サブシステムを提供する。

2

【０００５】
【課題を解決するための手段】前述した課題を実現するために、本発明は、記憶サブシステムにおいて、ジョブ多重の増加に伴い効果を発揮するジョブ管理、転送起動方式を提供する。
【０００６】データ転送を行うとき、ジョブ多重が発生すると１つのジョブ以外は全て転送資源待ち（待ち状態）となってしまい、転送終了時に次回転送起動ジョブ（最優先のジョブ）を待ち状態のジョブの中から見つけ出す必要がある。
【０００７】従来の技術では、ここで一斉にジョブを再起動していた。従って、ジョブの多重度が増せば増す程効率が悪くなり、時間も掛かってしまったが、本発明では、転送ジョブをキュー管理することでこれを解決することができる。本発明では、ジョブ起動時に転送資源が空いているかどうかをチェックし、空いていれば、当然資源を割り当てて転送を開始する。もし、空いていない場合は、転送資源待ちにする代わりに転送完了待ちとしてキューテーブルに次回転送の予約をする。転送完了待ちとすることでジョブでは、再起動を行う必要はない。しかし、実際は転送完了待ちどころか資源待ちであるため、現在転送中のジョブが割り込みの完了通知を行うタイミングで再起動～資源割り当ての作業を行うものとする。
【０００８】再起動～資源割り当ての作業を行う時、キューテーブルを参照することでキュー登録（予約）されている１ジョブのみの再起動を行い転送を開始することができる。キューテーブル参照時に、素早く目的のジョブが発見できる様に、キュー構造にはNEXTキューのアドレス情報を持たせ、キュー登録したジョブは、常に一つ前に登録したジョブキューに連結させる。したがって、転送完了割り込み時にキューテーブルを参照した際には、NEXTキューポインタが示す転送ジョブを起動することで効率的な連続転送が可能となる。
【０００９】
【発明の実施の形態】以下、本発明の実施例を図面により説明する。図１は、本発明の実現装置である記憶制御装置１０１とそれを繋いだホストコンピュータ１００との構成図である。本装置はチャネル側の処理を行うホストインタフェース制御装置１０２、ドライブ側の処理を行うドライブインタフェース制御装置１０９及び、不揮発化機構を有するキャッシュメモリ１０７、共用メモリ

１０６で構成し、各制御装置から信号線を介してキャッシュメモリ１０７、共用メモリ１０６へのデータ転送、制御を可能としている。

【００１０】記憶制御装置１０１では、まずホストコンピュータ１００の入出力データ要求に対しホストインタフェース制御装置１０２内の制御プロセッサ１０４が要求に応じて下部に処理を展開する。入出力データ要求によるデータ転送は、ホストインタフェース制御装置１０２内のコンフィグレーションメモリ空間１０３にセットされるエントリ情報に基づいて実行する。

【００１１】READコマンドに対しては、要求データをドライブインタフェース制御装置１０９を介してドライブ群（１１０～１１３）からキャッシュメモリ１０７へステージングし、キャッシュメモリ１０７にステージングされたデータはホストインタフェース制御装置１０２内の制御プロセッサ１０４から制御信号を受けたＤＭＡコントローラ１０５によりキャシュメモリ１０７からホストコンピュータ１００へのデータ転送を行う。またWRITEコマンドに対しては、要求データ分のキャッシュスロットをキャッシュメモリ１０７内に確保し、ホストインタフェース制御装置１０２内の制御プロセッサ１０４から制御信号を受けたＤＭＡコントローラ１０５がホストコンピュータ１００からキャッシュ１０７へのデータ転送を行い、さらにドライブインタフェース制御装置１０９を介してキャッシュ１０７からドライブ群（１１０～１１３）へ書き込まれる。

【００１２】図２、図３は本発明における動作原理の概要を説明する図である。以下図２、図３を用いて本発明の動作原理の概要を説明する。図２は、１転送ジョブの起動から終了までを示している。まず、ジョブ起動（２００）において実行可能となったジョブは転送資源（図１、DMAコントローラ１０５）が開放されていれば、転送を開始（２０１）させる。もし、別ジョブの実行により転送資源（ 図１、DMAコントローラ１０５）が空いていなければ、キューテーブル（図４で詳細を説明）に転送キューとして転送ジョブのエントリ情報を登録（２０２）する。ジョブの状態は、この時点で転送を開始した場合もキュー登録した場合も転送完了待ち（２０３）の待ち状態とする。転送完了待ち（２０３）となったジョブは 転送資源（図１、DMAコントローラ１０５）からの転送完了通知のプロセッサ割り込み後ジョブを終了させる（２０４）。

【００１３】転送完了通知割り込みからの処理については図３を使用して説明する。転送完了通知（３００）の延長の処理として割り込み制御プログラムで以下の処理を行う。まず、図２でキュー登録に使用したキューテーブルを参照（３０１）する。ここでは登録キューをサーチし、実行可能なキューを発見する（存在するなら）。実行可能なキューが存在するなら、完了通知を受けた現在のジョブキュー（この時点では削除されて存在してない）の次に登録されたジョブキューであり、キューテーブル上では論理的に前のジョブキューに繋がれたキューであることを意味する。このキューの転送ジョブは、キュー登録を行った時点で転送完了待ちの待ち状態となっており、この１ジョブのみを再起動（３０２）させる。起動したジョブキューは、直ちに削除する（３０３）。そして、起動したジョブに転送資源を割り当てて転送を開始（３０４）させる。転送開始後は、割り込み処理を終了させる（３０５）。一方、キューの参照（３０１）で実行可能なキューを発見できなかったなら、登録キュー無しにより同じく割り込み処理を終了させる（３０５）。

【００１４】以上が図２、図３で示す本発明のにおける動作原理の概要であるが、基本的に転送データの準備をジョブで完了していれば特にジョブで実行する必要はないと考え、本方式を実施している。割り込みの延長処理では未完了のデータがある限り、キューから情報を取り出し、転送実行を続けキューが空となった時点で処理を止める方式である。

【００１５】図４は、データ転送キューテーブルの構造を示す図である。キュー管理にはテーブル１、テーブル２の２つのテーブルを使用する。実際のジョブの情報を登録するデータ転送キューはテーブル２の方であり、テーブル１はこのテーブル２を管理する上で使用する。

【００１６】テーブル２のキュー構造は、当該キューの次に実行されるキューのアドレスを指すNEXTキューポインタ格納エリア（４０４）、データ入出力要求における１－Ｔ－Ｌ－Ｘ情報格納エリア（４０５）及び、転送ジョブのエントリ情報の格納アドレスを指す転送エントリポインタ格納エリア（４０６）からなる。また、テーブル１のキュー構造は、テーブル１における登録されたキューのうち先頭のキューアドレスを指す登録管理先頭ポインタ（４００）、同じく登録されたキューのうち最終のキューアドレスを指す登録管理最終ポインタ（４０１）、またテーブル１における空きキューのうち先頭のキューアドレスを指す空き管理先頭ポインタ（４０２）、同じく空きキューのうち最終のキューアドレスを指す空き管理最終ポインタ（４０３）からなる。

【００１７】図５から図９に、キュー操作一連の動作を示す。以下、各フローチャート図を用いてキュー操作の原理を説明する。図５、図６は、図４で示した２つのキューテーブルの初期化処理を示す。図５はテーブル１の初期化である。ステップ５００～ステップ５０３においてポインタ格納エリアに初期値をセットする。登録管理先頭ポインタ（４００）及び、登録管理最終ポインタ（４０１）には自ポインタアドレスを初期値に使用し、テーブル２に一つもキューが登録されていない状態示す。空き管理先頭ポインタ（４０２）及び、空キュー最終ポインタ（４０３）の初期値にはそれぞれテーブル２上の先頭キューアドレス、最終キューアドレスをセット

する。

【００１８】図６はテーブル２の初期化である。ステップ６００でテーブル２の先頭のキューをカレントキューとする。カレントキューが最終のキューを超えるまでステップ６０１の判定でステップ６０２へ移行し、カレントキューのテーブルアドレスを取得する。ステップ６０３でカレントキューが最終キューならカレントキューのNEXTキューポインタに最終キューアドレスを設定する（ステップ６０４）。ステップ６０３でカレントキューが最終キューでないならNEXTキューポインタにはテーブル上の次のキューアドレスをセットする（ステップ６０５）。この処理によって、カレントキューと次のキューが連結され、最終的にはすべてのキューがポインタで繋がれた形となる。次いで、ステップ６０６でカレントキューエリアを初期化（ゼロクリア）する。ステップ６０７で次のキューをカレントキューにセットしステップ６０１に戻る。ステップ６０１でカレントキューが最終キューを超えた時、初期化処理を終了する。

【００１９】図７は、キュー登録処理の動作を示す。キューの登録は、テーブル１（図４）における空き管理先頭ポインタ（４０２）が指すアドレス上のキューを登録対象のキュー（カレントキュー）として転送情報を登録するものである。まず、ステップ７００で登録キューの有り無しを判定する。もしすでに登録されたキューがあるならステップ７０２へ、もし登録されたキューが一つもないなら７０１へ移行し、フラグで識別する。ステップ７０３で空きキューの有り無しを判定する。キューが一つも空いていないならステップ７０８へ移行し、登録処理を終了する。キューが空いているならステップ７０４へ移行し、テーブル１（図４）から空き管理先頭ポインタ（４０２）を取得し、このポインタが指すアドレスのキューをカレントキューにする（ステップ７０５）。

【００２０】次いで、登録管理最終ポインタ（４０１）、カレントのNEXTキューポインタを取得する（ステップ７０６，ステップ７０７）。カレントキューが最後の空きキューなら（ステップ７０９）、空き管理先頭ポインタ（４０２）及び登録管理最終ポインタ（４０１）に自ポインタアドレスをセットし（ステップ７１１，７１２）、カレントキューが最後のキューでないなら、新たな空き管理先頭ポインタ（４０２）としてカレントキューのNEXTキューポインタが示すキューアドレスをセット（ステップ７１０）し、カレントキューには登録情報をセットする（ステップ７１３）。カレントのNEXTキューポインタには繋ぐ登録キューが存在しないので自ポインタアドレスをセットする（ステップ７１４）。

【００２１】次に、ステップ７００の登録キューの有り無し判定の結果、フラグがたっているなら（ステップ７１５）、カレントキューを一つ前に登録したキューに繋ぐ必要があるため（ステップ７１６）、一つ前のキューのNEXTキューポインタにカレントキューのテーブルアドレスをセットする。もし、ステップ７００の判定でフラグがたっていないなら（ステップ７１５）、カレントキューが登録キューの先頭であることを意味するので登録管理先頭ポインタ（４００）にカレントのキューアドレスをセットする（ステップ７１７）。キューアドレスをセットする（ステップ７１５）。最後に、新たな登録管理最終ポインタ（４０１）としてカレントのキューアドレスをセットする。

【００２２】図８は、キュー検索処理を示す。キューの検索では、テーブル１（図４）における登録管理先頭ポインタ（４００）が指すアドレス上のキューを実行対象のキュー（カレントキュー）として検索するものである。ステップ８００で登録キュー有り無し判定をする。キューが登録されていなければ、ステップ８０５に移行し検索処理を終了する。キューが登録されているなら、テーブル２（図４）から登録管理先頭ポインタ（４００）、登録管理最終ポインタ（４０１）を取得する（ステップ８０１，ステップ８０２）。

【００２３】次いで、最後に登録されたキューのNEXTキューポインタを取得し（ステップ８０３）、ステップ８０１で取得した登録管理先頭ポインタをカレントキューにセットする（ステップ８０４）。このカレントのキューポインタが最後に登録されたキューのNEXTキューポインタと一致しない間（ステップ８０６）、カレントキューが実行可能キューであるか判定する（ステップ８０７）。ここでの実行可否は、SCSIにおけるCA状態（Contingent allegiance）のチェックである、カレントキューのＩ－Ｔ－Ｌ－Ｘから当該ジョブがＣＡ状態でないかを確認する。カレントキューが実行可能であるなら、実行可能キュー通知を行う（ステップ８１１）。本通知を受けた割り込み処理により、カレントキューのジョブが再起動され、転送が始まる。実行可能キュー通知後は、直ちにカレントキューを削除し（ステップ８１２）、検索処理を終了する。

【００２４】一方、ステップ８０７の判定でカレントキューが実行可能キューでないと判断した場合は、カレントキューポインタを退避し（ステップ８０９）、次の登録キューをカレントキューにセットし、ステップ８０６の判定に戻る。この判定で、カレントのキューポインタが最後に登録されたキューのNEXTキューポインタと一致した時点で検索処理を終了する。

【００２５】図９は、キュー削除処理を示す。キューの削除は、キューの検索と一連の処理である。

【００２６】実行可能な登録キューを発見したなら、そのキューが削除処理のカレントキューとなる。

【００２７】まず、テーブル１から登録管理先頭ポインタ（４００）、登録最終ポインタ（４０１）を取得し、最後に登録されたキューのNEXTキューポインタを取得する（ステップ９００，ステップ９０１，ステップ９０２）。同じくテーブル１から空き管理先頭ポインタ（４

０２）、空き管理最終ポインタ（４０３）を取得する（ステップ９０３，ステップ９０４）。カレントキューが登録キューの先頭、または最終なら（ステップ９０５）、ステップ９０６へ移行し、もし登録されているキューが一つしかないなら空き管理先頭ポインタ（４０２）及び空きキュー終端ポインタ（４０３）に自ポインタアドレスをセットし（ステップ９１２，ステップ９１３）、カレントキュー以外にもキューが登録されていれば、ステップ９０７の判定へ移行する。

【００２８】カレントキューが登録されている先頭のキューなら、登録管理先頭ポインタ（４００）にカレントのNEXTキューポインタが示すキューアドレスをセットし（ステップ９１１）、ステップ９０７で先頭のキューでないなら、登録管理最終ポインタ（４０１）にひとつ前に登録した（図８のステップ８０９で退避した）キューアドレスをセットし、このキューのNEXTキューポインタにカレントキューのNEXTキューポインタが示すキューアドレスをセットする（ステップ９１０）。ステップ９０５の判定でカレントキューが先頭または最終でないならステップ９１０と同様にひとつ前に登録したキューのNEXTキューポインタにカレントキューのNEXTキューポインタが示すキューアドレスをセットする（ステップ９０８）。次にステップ９１４の判定で空きキューが無いなら、空き管理先頭ポインタ（４０２）にカレントのキューアドレスをセットし（ステップ９１６）、空きキューがあるなら、最終空きキューのNEXTポインタにカレントのキューアドレスをセット（ステップ９１５）し、ステップ９１７に移行する。次いで空き管理最終ポインタに（４０２）カレントのキューアドレスをセットし（ステップ９１７）、カレントキューは初期化（空きキューに）する。

【００２９】以上がキュー操作の原理であるが、テーブル２の登録キュー及び空きキューのNEXTキューポインタは、必ずしもテーブル上の次のキューアドレスを指し示しているわけではない。初期状態でのみ、すべて空きキューとしてそれぞれのキューのNEXTキューポインタは先頭から最終まで順番に次のキューを指している。しかし、ジョブが起動されて削除されたキューは、最終の空きキューの後に繋がれ、自キューが最終の空きキューとなり、また、キュー検索で実行されなかったカレントキューは実行可能になるまで登録キューのままであるため、このようにキュー登録、キュー検索、キュー削除が頻繁に行われるうちに空きキュー、登録キューのテーブル上の並びはばらばらであり、登録可能な順番もしくは登録された順番ではない。各キューのNEXTキューポインタは、テーブル上不規則に並んだキューの中で自キューの次に登録可能なキュー、もしくは自キューの次に実行可能なキューを正確に示しており、テーブル１（図４）がその先頭と最終を管理することですべてのキュー操作を円滑に行えるしくみとなっている。

【００３０】

【発明の効果】以上の実施例に示したように、本発明によればｌジョブの転送完了時には、次回転送のジョブがキューを参照することにより、即時に明確になるため不要なジョブを起動することなく、目的のジョブのみの起動ができる。これにより、転送と転送の継ぎ目に要するオーバヘッドを最小限にすることができ、プロセッサを効率的に使用できる。これは、ジョブの多重度が増せば増すほど効果を発揮する。

【図面の簡単な説明】

【図１】本発明の実施例における記憶サブシステムの構成を示す図である。

【図２】本発明によるジョブの動作原理を説明するための図である。

【図３】本発明による割り込み処理の動作原理を説明するための図である。

【図４】本発明によるデータ転送キュー管理テーブルの構成を示す図である。

【図５】本実施例におけるキュー操作（キューテーブルの初期化）のフローチャートである。

【図６】本実施例におけるキュー操作（キューテーブルの初期化）のフローチャートである。

【図７】本実施例におけるキュー操作（キューの登録）のフローチャートである。

【図８】本実施例におけるキュー操作（キューの検索）のフローチャートである。

【図９】本実施例におけるキュー操作（キューの削除）のフローチャートである。

【符号の説明】

１００　ホストコンピュータ　　１０１　記憶制御装置

１０２　ホストインタフェース制御装置

１０３　コンフィグレーションメモリ空間

１０４　制御プロセッサ　　１０５　DMAコントローラ

１０６　共用メモリ　　１０７　キャッシュメモリ

１０８　信号線　　１０９　ドライブインタフェース制御装置

１１０～１１３　ドライブ

４００　登録管理先頭ポインタ　　４０１　登録管理最終ポインタ

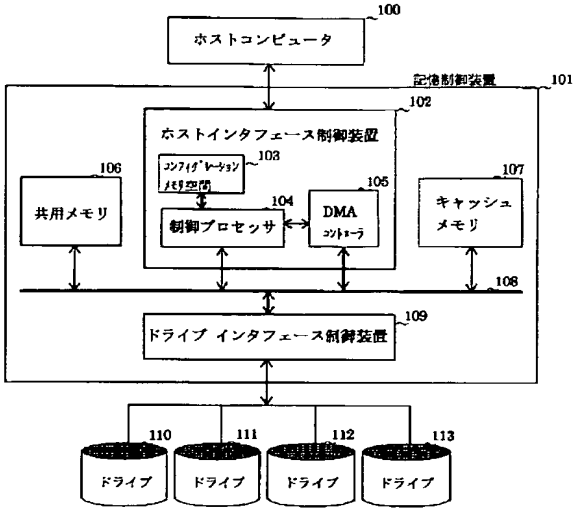
４０２　空き管理先頭ポインタ　　４０３　空き管理最終ポインタ

４０４　NEXTキューポインタ　　４０５　Ｉ－Ｔ－Ｌ－Ｘ

４０６　転送エントリポインタ

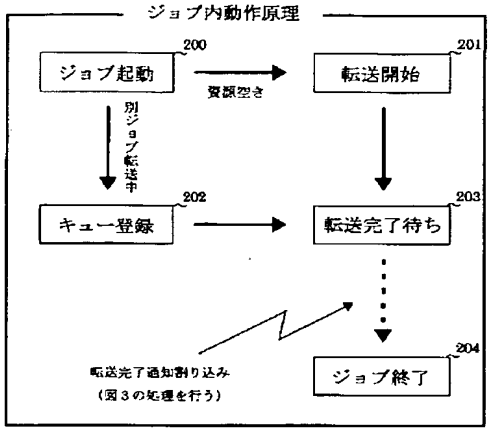
【図１】

図１



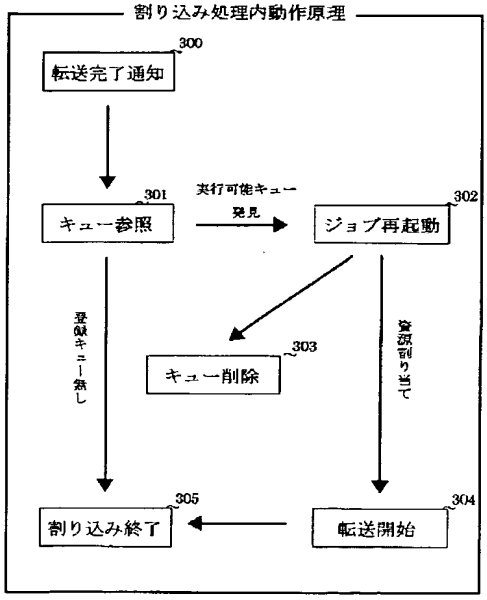

【図２】

図２



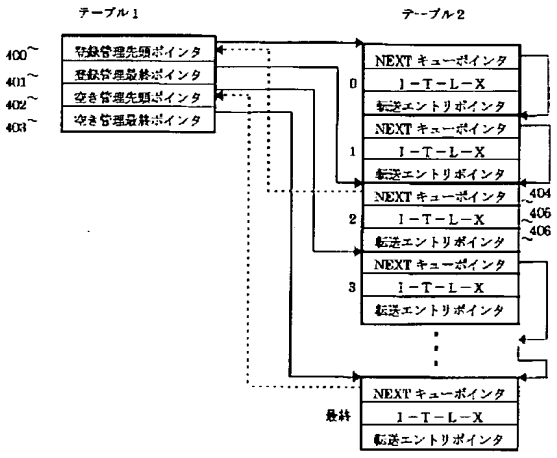

【図３】

図３



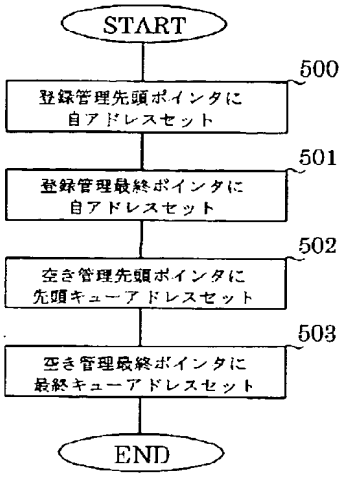

【図４】

図４



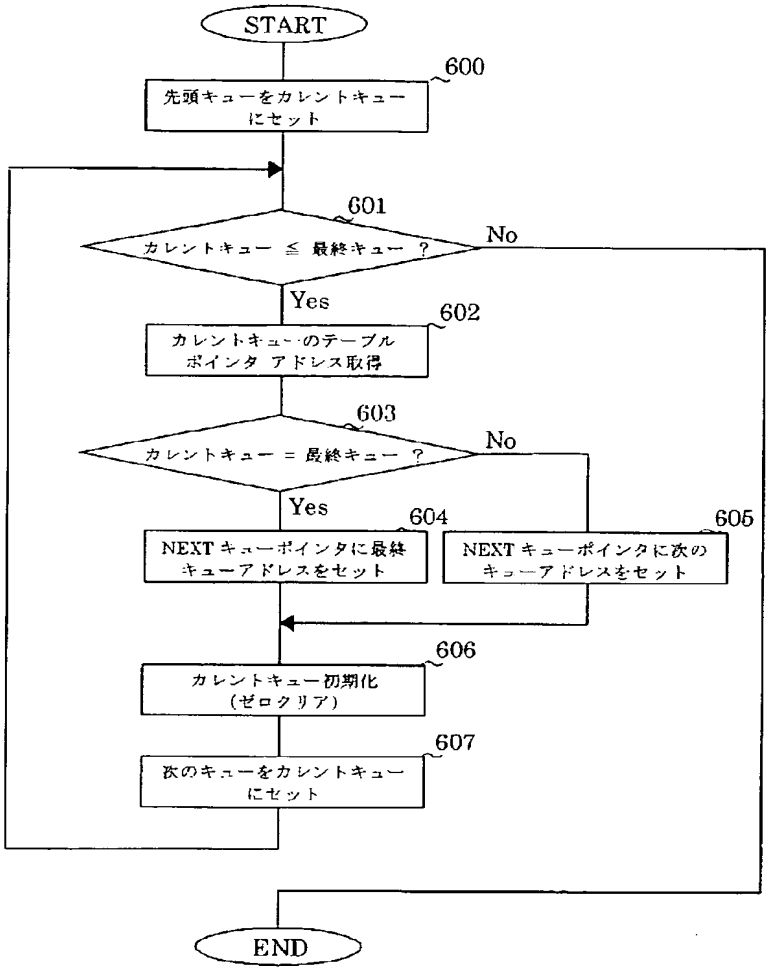
【図５】

図５


START
500
登録管理先頭ポインタに
自アドレスセット
501
登録管理最終ポインタに
自アドレスセット
502
空き管理先頭ポインタに
先頭キューアドレスセット
503
空き管理最終ポインタに
最終キューアドレスセット
END

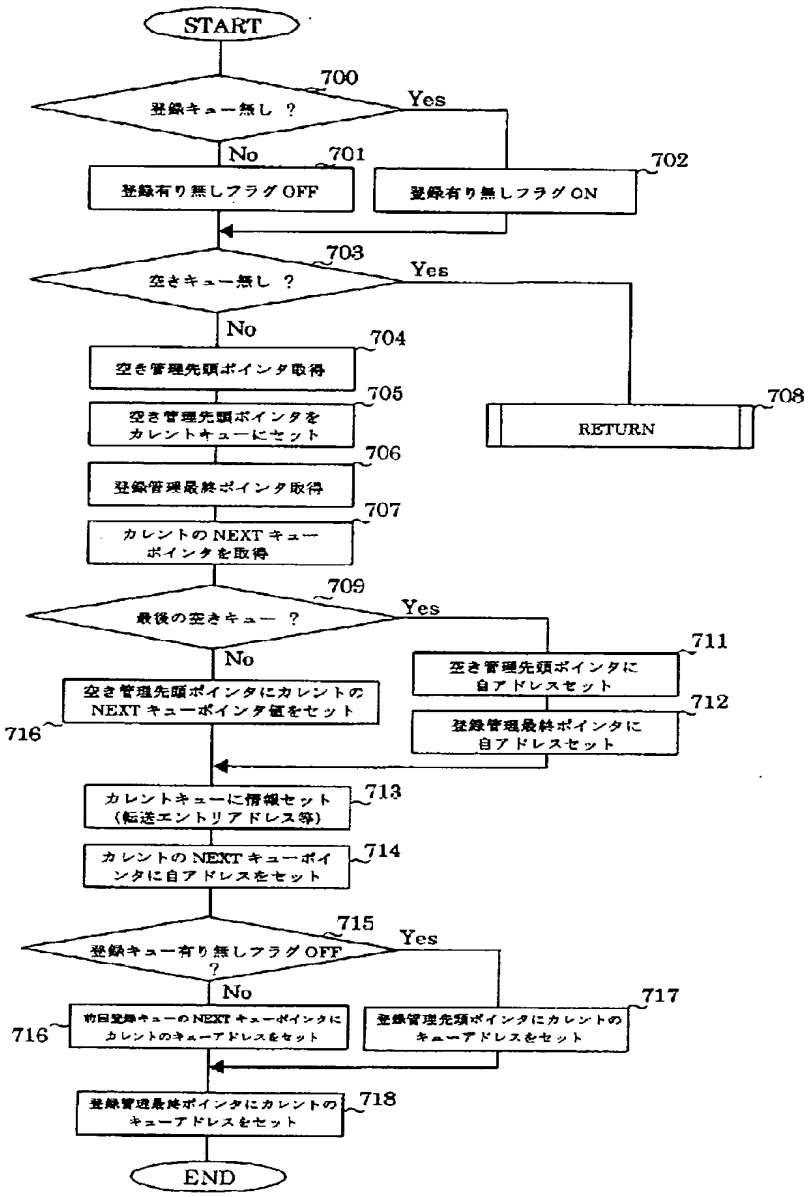
【図６】

図６


START
600
先頭キューをカレントキュー
にセット
601
カレントキュー ≦ 最終キュー ？
No
Yes
602
カレントキューのテーブル
ポインタ アドレス取得
603
カレントキュー = 最終キュー ？
No
Yes
604
NEXT キューポインタに最終
キューアドレスをセット
605
NEXT キューポインタに次の
キューアドレスをセット
606
カレントキュー初期化
（ゼロクリア）
607
次のキューをカレントキュー
にセット
END

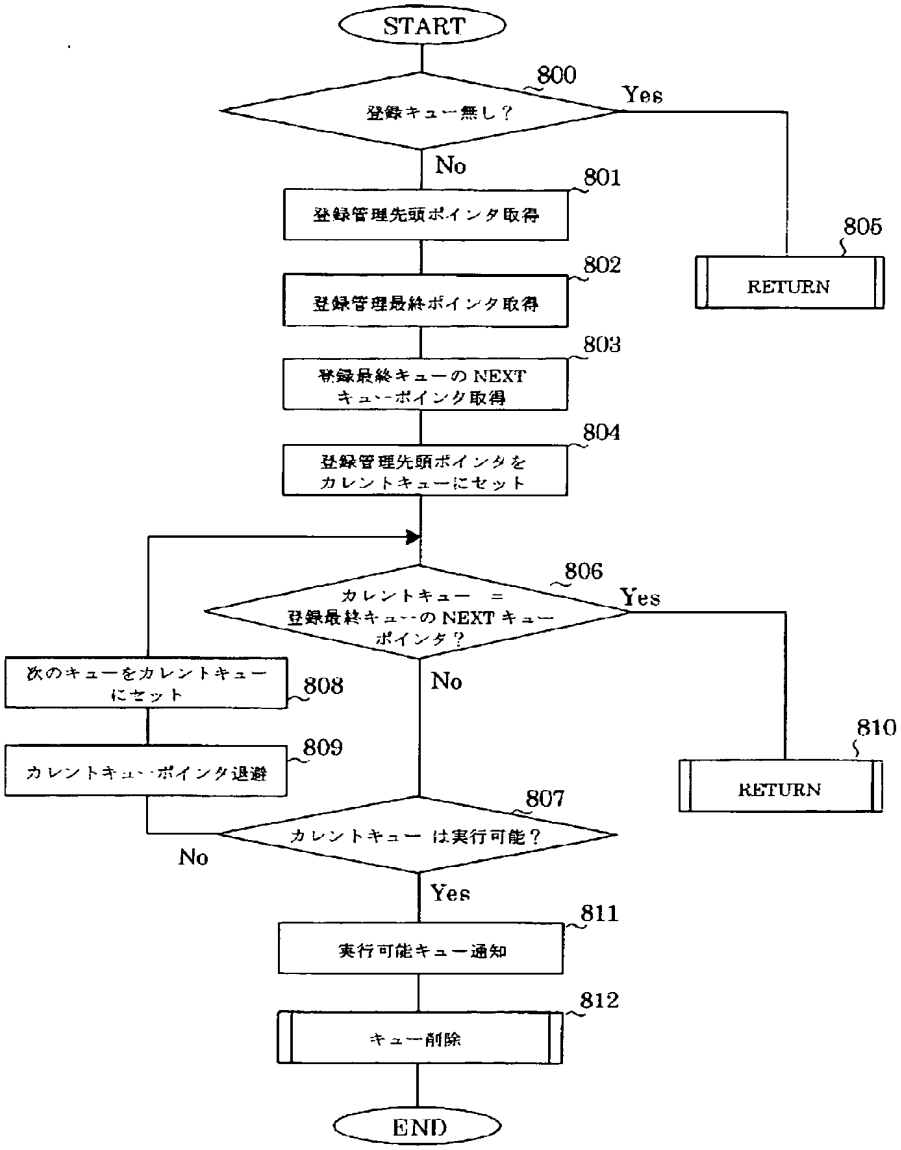
【図７】

図７


START
700
登録キュー無し ？
Yes
No
701
登録有り無しフラグ OFF
702
登録有り無しフラグ ON
703
空きキュー無し ？
Yes
No
704
空き管理先頭ポインタ取得
705
空き管理先頭ポインタを
カレントキューにセット
706
登録管理最終ポインタ取得
707
カレントの NEXT キュー
ポインタを取得
708
RETURN
709
最後の空きキュー ？
Yes
No
711
空き管理先頭ポインタに
自アドレスセット
712
登録管理最終ポインタに
自アドレスセット
716
空き管理先頭ポインタにカレントの
NEXT キューポインタ値をセット
713
カレントキューに情報セット
（転送エントリアドレス等）
714
カレントの NEXT キューポイ
ンタに自アドレスをセット
715
登録キュー有り無しフラグ OFF
？
Yes
No
716
前回登録キューの NEXT キューポインタに
カレントのキューアドレスをセット
717
登録管理先頭ポインタにカレントの
キューアドレスをセット
718
登録管理最終ポインタにカレントの
キューアドレスをセット
END

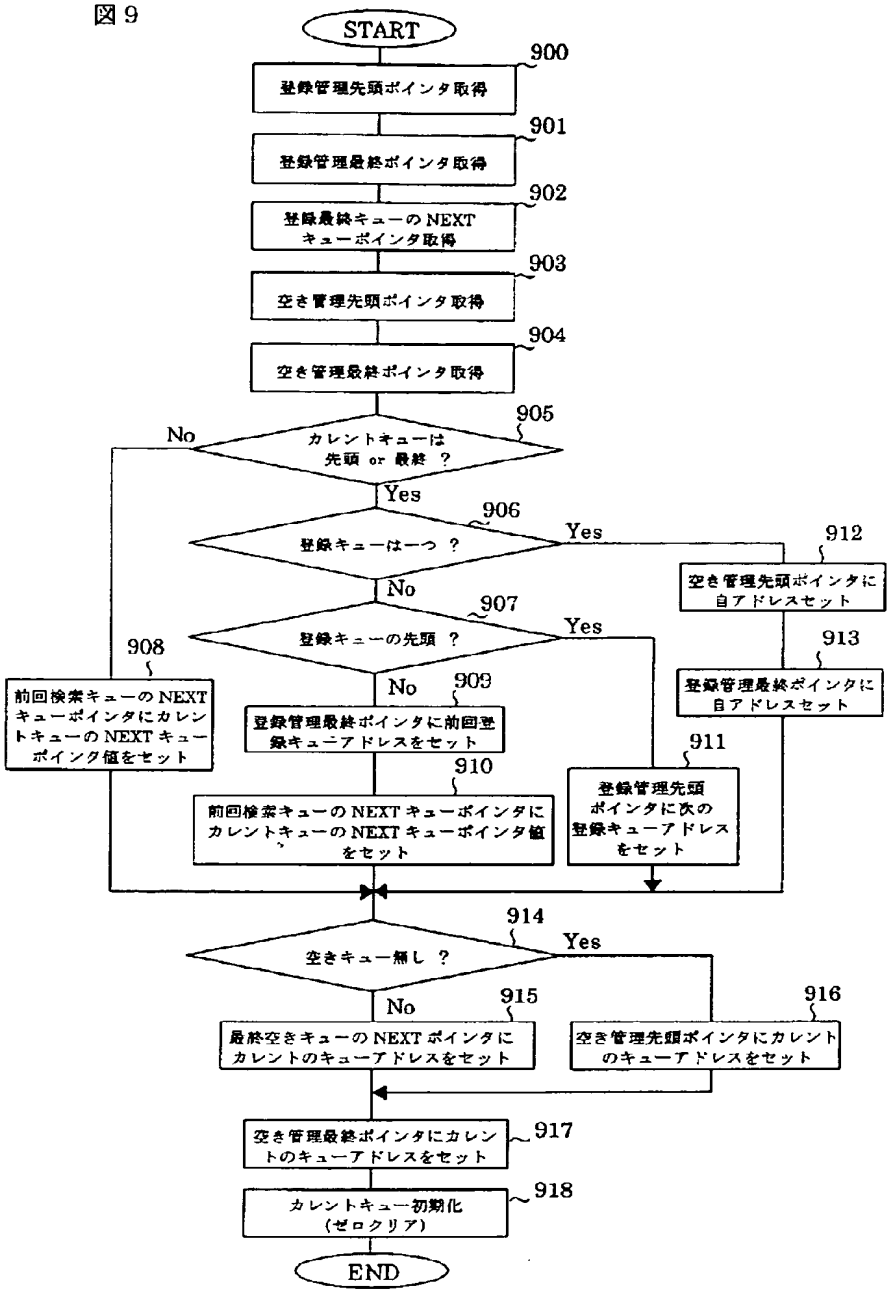
【図８】

図８


START
800
登録キュー無し？
Yes
No
801
登録管理先頭ポインタ取得
802
登録管理最終ポインタ取得
803
登録最終キューのNEXTキューポインタ取得
804
登録管理先頭ポインタをカレントキューにセット
805
RETURN
806
カレントキュー＝登録最終キューのNEXTキューポインタ？
Yes
No
808
次のキューをカレントキューにセット
809
カレントキューポインタ退避
810
RETURN
807
カレントキューは実行可能？
No
Yes
811
実行可能キュー通知
812
キュー削除
END

【図９】

図９


START
900
登録管理先頭ポインタ取得
901
登録管理最終ポインタ取得
902
登録最終キューのNEXTキューポインタ取得
903
空き管理先頭ポインタ取得
904
空き管理最終ポインタ取得
905
カレントキューは先頭 or 最終 ？
No
Yes
906
登録キューは一つ ？
Yes
912
空き管理先頭ポインタに自アドレスセット
No
907
登録キューの先頭 ？
Yes
913
登録管理最終ポインタに自アドレスセット
908
前回検索キューのNEXTキューポインタにカレントキューのNEXTキューポインタ値をセット
No
909
登録管理最終ポインタに前回登録キューアドレスをセット
911
登録管理先頭ポインタに次の登録キューアドレスをセット
910
前回検索キューのNEXTキューポインタにカレントキューのNEXTキューポインタ値をセット
914
空きキュー無し ？
Yes
No
915
最終空きキューのNEXTポインタにカレントのキューアドレスをセット
916
空き管理先頭ポインタにカレントのキューアドレスをセット
917
空き管理最終ポインタにカレントのキューアドレスをセット
918
カレントキュー初期化（ゼロクリア）
END

フロントページの続き

(72)発明者　桑原　宏
神奈川県小田原市国府津2880番地　株式会
社日立製作所ストレージシステム事業部内